

MongoDB教程

版本：1.0.0

更新时间：2025-04-13

作者：旭日彬彬

学习难度：简单

基础篇

简介

MongoDB 是一个开源的 NoSQL **文档数据库**。它属于非关系型数据库中的文档存储类型，使用类似 JSON 的 BSON 格式存储数据。数据以 **文档(Document)** 形式存储，而文档组织在 **集合(Collection)** 中，而集合组织在 **数据库(Database)** 中。

基本概念

在学习这个数据库之前，我们先了解一下他的基本概念：**数据库**，**集合**，**文档**。

数据库：是用于存放**集合**（Collection）的**容器**。请把他简单的看作一个容器，没有其他的概念，就是一个容器。MongoDB中可以创建多个数据库。

集合：用于存放**文档**（Document）的**容器**。没错，也请把他简单的看作一个容器。

文档：MongoDB数据库存储数据的最小组织单元，是一个类JSON结构的文档，我们把他称为BSON。对JSON结构不了解的读者可参考链接：<https://www.runoob.com/json/json-tutorial.html>

连接数据库

连接数据库的工具有很多，我们采用官方的Mongo Shell作为演示，他是一个命令程序。官网下载链接：<https://www.mongodb.com/try/download/shell>

安装过程一路点Next，傻瓜式安装即可。

安装完成后，**mongosh**命令会自动注册到电脑中。终端中输入下面的命令查看版本，判断软件是否安装成功。找不到命令可尝试重启电脑。

```
mongosh --version
```

下面开始连接MongoDB，连接语法如下

```
mongosh mongodb://[username:password@]hostname:port[/database]
```

示例图片如下：

```
PS C:\Users\xurib\Desktop> mongosh mongodb://192.168.1.100:27017
Current Mongosh Log ID: 67f8e19f929da5f92b5f898
Connecting to:   mongodb://<credentials>@47.113.148.241:27017/?directConnection=true&appName=mongosh+2.5.0
Using MongoDB:   8.0.0
Using Mongosh:   2.5.0
```

数据库操作

前提已连接到MongoDB数据库，这里不在演示

常用命令如下：

db	//查看当前所在数据库
show dbs	//查看所有不为空的数据库，空的数据库不会显示
use database	//切换到指定数据库，不存在则创建
db.dropDatabase()	//删除当前所在的数据库

综合示例如下：

```
test> show dbs          查看所有不为空的数据库，空的数据库不会显示
admin   180.00 KiB
config 108.00 KiB
local   72.00 KiB
test> db               查看当前所在的数据库，连接MongoDB时，若没有指定数据库，则默认连接到 test 数据库
test
test> use test02        切换到 test02 数据库，若没有则创建该数据库，但由于此时数据库为空，故打印所有数据库时不显示
switched to db test02
test02> db
test02
test02> show dbs
admin   180.00 KiB
config 108.00 KiB
local   72.00 KiB
test02> use alibaba
switched to db alibaba
alibaba> show dbs
admin   180.00 KiB
config 108.00 KiB
local   72.00 KiB
alibaba> db.dropdatabase; 删除数据库，图中为错误语法，应为 db.dropDatabase()
alibaba.dropdatabase
```

集合操作

前提已连接到MongoDB数据库，这里不在演示

常用命令如下：

```
db.createCollection("name")    //创建集合
db.collection.drop()           //删除集合
show collections               //查看当前数据库所有集合
show tables                    //查看当前数据库所有集合
```

文档操作

前提已连接到MongoDB数据库，这里不在演示

常用命令如下：

```
//插入单个文档
db.collection.insertOne(document, [option])
//插入多个文档
db.collection.insertMany(document, [option])
//更新单个文档
db.collection.updateOne(filter, update, [options])
//更新多个文档
db.collection.updateMany(filter, update, [options])
//替换为新文档
db.collection.replaceOne(filter, newdoc, [options])
//查找并更新文档然后返回文档，可设置返回更新前或之后的文档
db.collection.findOneAndUpdate(filter, update, [options])
```

综合示例如下：

注意：

- option 参数对象是可选的，用于对插入操作提供额外控制，参考值如下：

参数	类型	说明
ordered	bool	是否无序插入
upsert	bool	如果没有匹配的文档，是否插入一个新文档
arrayFilters		当更新嵌套数组时，指定应更新的数组元素的条件
collation		指定比较字符串时使用的排序规则
returnDocument	string	固定值before/after。findOneAndUpdate中可用，返回旧或新的文档

- 如果集合定义了验证规则，插入文档时会进行验证，如果不符合规则，插入会失败。

删除文档

常用命令如下：

```
//删除单个文档
db.collection.deleteOne(filter, [options])
//删除多个文档
db.collection.deleteMany(filter, [options])
//查找并删除单个文档
db.collection.findOneAndDelete(filter, [options])
```

综合示例如下：

注意：

options 参数对象是可选的，用于对删除操作提供额外的控制，参考如下：

参数	类型	说明
writeConcern		指定写操作的确认级别
collation		指定比较字符串时使用的排序规则
projection	object	findOneAndDelete中可用，指定返回的字段
sort		findOneAndDelete中可用，指定排序顺序以确定要删除的文档

查询文档

常用命令如下：

```
//
```